ФПМИ БГУ

Вычислительные методы алгебры
Лаборатоная работа 8

Нахождение собственных значений и собственных векторов методом Данилевского

Подготовил:
Ткачук Павел
2 курс 1 группа

Преподаватель:
Будник Анатолий Михайлович

20 декабря 2016 г.

1 Постановка задачи

Входные данные:

$$A_{\text{исх.}} = \begin{bmatrix} 0.6444 & 0.0000 & -0.1683 & 0.1184 & 0.1973 \\ -0.0395 & 0.4208 & 0.0000 & -0.0802 & 0.0263 \\ 0.0132 & -0.1184 & 0.7627 & 0.0145 & 0.0460 \\ 0.0395 & 0.0000 & -0.0960 & 0.7627 & 0.0000 \\ 0.0263 & -0.0395 & 0.1907 & -0.0158 & 0.5523 \end{bmatrix}$$

Для того чтобы все собственные значения были действительными числами, домножим слева исходную матрицу на ей транспонированную:

$$A = A_{\text{исх.}} A_{\text{исх.}}^T = \begin{bmatrix} 0.4965221 & -0.02976049 & -0.10906373 & 0.13191428 & 0.09195098 \\ -0.02976049 & 0.18575662 & -0.05029722 & -0.06272879 & -0.0018678 \\ -0.10906373 & -0.05029722 & 0.59823034 & -0.06163865 & 0.17564755 \\ 0.13191428 & -0.06272879 & -0.06163865 & 0.59248754 & -0.02931901 \\ 0.09195098 & -0.0018678 & 0.17564755 & -0.02931901 & 0.34390336 \end{bmatrix}$$

Задача:

1. Для данной матрицы A вычислить коэфициенты $p_1, p_2, \ldots, p_n$ собственного многочлена

$$P(A) = \lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - \ldots - p_{n-1}\lambda - p_n$$

   (проверить совпадает ли $p_1 = Sp(A)$ и $p_n = \det(A)$)

2. По полученному характеристическому многочлену вычислить максимальное собственное значение λ_{max} матрицы

3. Вычислить собственный вектор $\vec{v}$ соответствующий данному значению

4. Найти невязку $\vec{r} = A\vec{v} - \lambda_{max}\vec{v}$

2 Алгоритм

1. Находим нормальную форму Фробениуса Φ матрицы A

$$A_n = A$$
$$A_1 = \Phi$$
$$A_i = M_i^{-1} A_{i+1} M_i,\ i = n-1, \ldots, 1,\ \text{где}$$

$$M_i = \begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ \end{array} \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ & & & \cdots & & & \\ -\frac{a_{i+1,1}^{i-1}}{a_{i+1,i}^{i-1}} & \cdots & -\frac{a_{i+1,i-1}^{i-1}}{a_{i+1,i}^{i-1}} & \frac{1}{a_{i+1,i}^{i-1}} & -\frac{a_{i+1,i+1}^{i-1}}{a_{i+1,i}^{i-1}} & \cdots & -\frac{a_{i+1,n}^{i-1}}{a_{i+1,i}^{i-1}} \\ & & & \cdots & & & \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix},$$

$$M_i^{-1} = \begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ \end{array} \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ & & & \cdots & & & \\ a_{i+1,1} & \cdots & a_{i+1,i-1}^{i-1} & a_{i+1,i}^{i-1} & a_{i+1,i+1}^{i-1} & \cdots & a_{i+1,n}^{i-1} \\ & & & \cdots & & & \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

2. Первая строка матрицы Φ представляет собой коэфициенты $(p_1, p_2, \ldots, p_n)$ характеристического многочлена исходной матрицы

3. Находим λ_{max} - макисмальный корень характеристического уравнения

$$\lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - \ldots - p_{n-1}\lambda - p_n = 0$$

4. По полученному λ_{max} вычисляем собственный вектор, соотвествующий данному значению

$$\begin{aligned} &\vec{v} = Sy, \text{ где} \\ &S = M_{n-1}M_{n-2}\ldots M_1, \\ &y = (\lambda_{max}^{n-1}, \ldots, \lambda_{max}, 1)^T. \end{aligned}$$

3 Результаты и вывод

3.1 Входные данные

0.6444 0.0000 -0.1683 0.1184 0.1973
-0.0395 0.4208 0.0000 -0.0802 0.0263
0.0132 -0.1184 0.7627 0.0145 0.0460
0.0395 0.0000 -0.0960 0.7627 0.0000
0.0263 -0.0395 0.1907 -0.0158 0.5523

3.2 Выходные данные

```
Ищем собственные вектор матрицы
Исходная матрица:
[[ 0.4965221  -0.02976049 -0.10906373  0.13191428  0.09195098]
 [-0.02976049  0.18575662 -0.05029722 -0.06272879 -0.0018678 ]
 [-0.10906373 -0.05029722  0.59823034 -0.06163865  0.17564755]
 [ 0.13191428 -0.06272879 -0.06163865  0.59248754 -0.02931901]
 [ 0.09195098 -0.0018678   0.17564755 -0.02931901  0.34390336]]
Коэфициенты характеристического многочлена
 [ 2.21689996 -1.82259145  0.68304846 -0.11469321  0.0069546 ]
q_1-Sp(A) =  3.10862446895e-15 q_n-det(A) =  7.12103986888e-16
Собственное значение:  0.78083494486
Собственный вектор:  [-2.19705047  0.13110161  3.17550372 -2.77730903  1.        ]
Невязка:  [ -7.79376563e-14   2.07611706e-14  -3.13082893e-13  -2.10320650e-12
  -6.66133815e-16]
Норма невязки:  2.12791076379e-12
```

3.3 Вывод

Данный метод относится к точным методам и позволяет решать проблему собственных значений. С его помощью можно находить точное решение данной проблемы, единственные погрешности которые возникают, возникают за счёт неточности машинных вычислений.

Сравним данный метод вычисления собственных векторов с методом Крылова, рассмотренном в предыдущей работе. Рассмотрим коэфициенты характерестического многочлена, полученные в обоих методах:

```
[ 2.21689996 -1.82259145  0.68304846 -0.11469321  0.0069546 ] - метод Крылова
[ 2.21689996 -1.82259145  0.68304846 -0.11469321  0.0069546 ] - метод Данилевского
```

Как видим, коэфициенты совпадают, значит будут совпадать и собственные значения.

Сравним собственные вектора из обоих методов(предварительно их пронормировав)

```
[-2.19705047  0.13110161  3.17550372 -2.77730903  1.        ] - метод Крылова
[-2.19705047  0.13110161  3.17550372 -2.77730903  1.        ] - метод Данилевского
```

Собственные вектора также, совпали однако можно заметить что нормы невязок этих векторов отличаються на несколько порядков, это объясняется тем, что норма собственного вектора полученного в методе Крылова

```
 [ 0.00382651 -0.00022833 -0.00553064  0.00483712 -0.00174166]
```

мала сама по себе, несмотря на это количество верных цифр в обоих методах одинаково.

4 Листинг кода

```
import numpy as np
import numpy.linalg as linalg


# Поиск собственных векторов методом Данилевского


def eigen(matr_A):
    A = matr_A.copy()
    size = len(A)
    S = np.eye(size)
    for i in reversed(range(0, size-1)):
        M = np.eye(size)
        invM = np.eye(size)
        M[i] = [- A[i+1, j] / A[i+1, i] for j in range(size)]
        M[i, i] = 1 / A[i+1, i]
        invM[i] = [A[i+1, j] for j in range(size)]
        A = np.dot(invM, A)
        A = np.dot(A, M)
        S = np.dot(S, M)

    print("Коэфициенты характеристического многочлена\n", A[i])
    print("q_1-Sp(A) = ", abs(matr_A.trace() - A[0, 0]), "q_n-det(A) = " ,abs(linalg.det(matr_A) - A
    p = [1] + list(A[i] * -1)  # Получили коэфициенты собственного многочлена
    eigvals = np.roots(p)
    y = [eigvals[0] ** (size - i - 1) for i in range(size)]
    return np.dot(S, y), eigvals[0]

file = open("matrix", "r")  # Чтение файла
A, b = [], []
for line in file:
    A.append([float(el) for el in line.split()[:-1]])
    b.append(float(line.split().pop()))
A = np.array(A)
A = np.dot(A, A.transpose())
b = np.array(b)
print("Ищем собственные вектор матрицы")
print("Исходная матрица:")
print(A)
ans = eigen(A)
print("Собственное значение: ", ans[1])
print("Собственный вектор: ", ans[0])
print("Невязка: ", np.dot(A, ans[0]) - ans[0] * ans[1])
print("Норма невязки: ", linalg.norm(np.dot(A, ans[0]) - ans[0] * ans[1]))
```